SYSTEM ROM USER'S GUIDE

The System ROM consists of a collection of routines which are entered by a hardware interrupt, a software interrupt, or system reset. The purpose of these routines is to initialize the system at power-up and provide a consistent interface to the hardware elements of the system.

The following table summarizes the interrupts by number, name, type, and address:

INTERRUPT	ADDRESS	TYPE	INTERRUPT NAME
0	0000:0000	HW	Zerodivide
1	0000:0004	HW	Single Step
2	0000:0008	HW	Non-Maskable Interrupt
3	0000:000C	HW	Software Breakpoint
4	0000:0010	HW	Arithmetic Overflow
5	0000:0014	SW	Print Screen
6	0000:0018	SW	Not Used
7	0000:001C	SW	Not Used
8	0000:0020	HW	IRQ 0 - Timer Interrupt
9	0000:0024	HW	IRQ 1 - Keyboard Interrupt
A	0000:0028	HW	IRQ 2 - Not Used
B	0000:002C	HW	IRQ 3 - Communications, Secondary
C	0000:0030	HW	IRQ 4 - Communications, Primary
D	0000:0034	HW	IRQ 5 - Not Used
E	0000:0038	HW	IRQ 6 - Diskette Interrupt
F	0000:003C	HW	IRQ 7 - Printer Interrupt
10	0000:0040	SW	Video I/O
11	0000:0044	SW	Equipment Configuration
12	0000:0048	SW	Memory Sizes
13	0000:004C	SW	Diskette I/O
14	0000:0050	SW	Communications (RS-232) I/O
15	0000:0054	SW	Not Used
16	0000:0058	SW	Keyboard I/O
17	0000:005C	SW	Printer I/O
18	0000:0060	SW	Not Used
19	0000:0064	SW	Bootstrap
1A	0000:0068	SW	Time of Day I/O
1B	0000:006C	SW	CTRL-BREAK Vector
1C	0000:0070	SW	Timer Vector
1D	0000:0074	SW	Video Init Parameter Table Vector
1E	0000:0078	SW	Diskette Parameter Table Vector
IF	0000:007C	SW	Graphics Dot Table Vector

Additionally, there is also information on special fixed locations in the System ROM:

NAME	ADDRESS	LENGTH	CONTENTS
REVISION	F000:FFE6	4	ROM revision in ASCII
MACHINE ID	F000:FFEA	6	`COMPAQ' in ASCII

INTERRUPT: 0

LOCATION: 0000:0000

NAME: ZERODIVIDE

DESCRIPTION: This interrupt is initiated by the 8088 microprocessor when a Divide by Zero or Divide Overflow takes place.

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

INPUTS: None.

OUTPUTS: None.

USE: It is the responsibility of DOS or the applications program to set up the vector to intercept 8088 DIV and IDIV instruction exceptions.

INTERRUPT: 1

LOCATION: 0000:0004

NAME: SINGLE STEP

DESCRIPTION: This interrupt is initiated by the 8088 microprocessor when an instruction is executed with the trace flag (TF) set.

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

INPUTS: None.

OUTPUTS: None.

USE: It is the responsibility of DOS or the applications program (usually DEBUG or DDT) to set up this vector for useful action.

INTERRUPT: 2

LOCATION: 0000:0008

NAME: NON-MASKABLE INTERRUPT

DESCRIPTION: This interrupt is initiated by hardware external to the 8088 microprocessor to process conditions caused by one of three sources:

1) Parity error on the processor board.
2) Parity error from the I/O channel.
3) 8087 interrupt.

INITIALIZED: Points to ROM INT 2 handler.

ROM ACTION: When a Non-Maskable Interrupt occurs, the status of the parity error hardware latches is examined. For parity errors on the processor board, it displays "PARITY CHECK 1", then halts. For parity errors from the I/O channel, it displays "PARITY CHECK 2", then halts. If the cause of the interrupt is the 8087 (or explicit software INT 2 call), then control simply returns.

INPUTS: State of the hardware parity error latches.

OUTPUTS: Video display (if parity error).

USE: This vector may be set up to handle the 8087 by saving the contents of location 0000:0008 and loading the location with a pointer to the user service routine. This routine can determine if the 8087 interrupted by storing and examining the contents of the 8087 status register. Return can then be through an indirect FAR JMP using the contents of the saved location (to handle those interrupts not caused by the 8087).

INTERRUPT: 3

LOCATION: 0000:000C

NAME: SOFTWARE BREAKPOINT

DESCRIPTION: This interrupt is initiated by execution of an INT 3 instruction (opcode CCh).

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

INPUTS: None.

OUTPUTS: None.

USE: It is the responsibility of DOS or the applications program (usually DEBUG or DDT) to set up this vector for useful action.

INTERRUPT: 4

LOCATION: 0000:0010

NAME: ARITHMETIC OVERFLOW

DESCRIPTION: This interrupt is initiated by execution of an INTO instruction when the overflow flag (OF) is set.

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

INPUTS: None.

OUTPUTS: None.

USE: It is the responsibility of DOS or the applications program to set up the vector to process this interrupt.

INTERRUPT: 5

LOCATION: 0000:0014

NAME: PRINT SCREEN

DESCRIPTION: This interrupt is initiated by execution of an INT 5 instruction and is used to obtain a hard-copy of the video display screen.

INITIALIZED: Points to ROM Print Screen handler.

ROM ACTION: It is normally called by the ROM keyboard handler when SHIFT+PRTSC is typed. This interrupt in turn uses INT 10h to read the screen and INT 17h to send characters to the printer.

The cursor position at the time this routine is called is saved and restored when the printing has completed. Address 0050:0000 contains the status of the print screen operation; 0 = PRINT SCREEN NOT IN OPERATION (or successful completion of a print screen call); 1 = PRINT SCREEN IS IN PROGRESS. Another value means an error was encountered during a print screen operation.

If another print screen call is executed while a print screen is in progress, it will be ignored. Trailing spaces are not compressed. All registers are preserved. This routine runs with interrupts enabled.

INPUTS: None.

OUTPUTS: 0050:0000 = status of print screen.

USE: An applications program may request it. This vector is normally not changed by the user.

INTERRUPT: 6

LOCATION:	0000:0018
NAME:	--
DESCRIPTION:	Not used.
INITIALIZED:	0000:0000
ROM ACTION:	None.

INTERRUPT: 7

LOCATION:	0000:001C
NAME:	--
DESCRIPTION:	Not used.
INITIALIZED:	0000:0000
ROM ACTION:	None.

INTERRUPT: 8

LOCATION: 0000:0020

NAME: IRQ 0 - TIMER INTERRUPT

DESCRIPTION: This interrupt is hardware-initiated by the output of Counter 0 of the 8253 Programmable Interval Timer at a rate of 18.2 times per second (18.2 Hz). It is used to provide timekeeping functions, turn off the disk motors, and make calls to INT 1Ch.

INITIALIZED: Points to ROM Timer Interrupt handler.

ROM ACTION: ROM code increments the contents of a 32-bit double word. When the count reaches 1573040 (1800B0h), a flag is set to a 1 to indicate that the timer has rolled past a day since last read and the 32-bit double word is cleared to 0 for the next day's incrementing.

INPUTS: Double word counter.

OUTPUTS: Double word counter.
Rolled-over flag.

USE: The contents of the counter may be set or read by calls to INT 1Ah and used to keep track of elapsed time (since midnight). A user-supplied routine may also be periodically invoked from INT 1Ch. See INTERRUPT 1A and INTERRUPT 1C for details.

This vector is normally not changed by the user.

INTERRUPT: 9

LOCATION: 0000:0024

NAME: IRQ 1 - KEYBOARD INTERRUPT

DESCRIPTION: This is a hardware interrupt which occurs each time a key is hit or released at the keyboard.

INITIALIZED: Points to ROM Keyboard Interrupt handler.

ROM ACTION: The interrupt routine reads the key from the keyboard registers, encodes the key or takes special action if required, notifies the keyboard the key has been read, clears the 8259A Interrupt Controller, and loads the encoded key into the keyboard buffer.

INPUTS: Keyboard.

OUTPUTS: Keyboard FIFO buffer.

USE: This vector is normally not changed by the user.

INTERRUPT: A

LOCATION: 0000:0028

NAME: IRQ 2

DESCRIPTION: Not used.

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

INPUTS: None.

OUTPUTS: None.

USE: If a user-supplied, interrupt-driven device uses the IRQ2 interrupt line, then location 0000:0028 is loaded with the address of the interrupt service routine.

INTERRUPT: B

LOCATION: 0000:002C

NAME: IRQ 3 - COMMUNICATIONS, SECONDARY

DESCRIPTION: Not used.

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

INPUTS: None.

OUTPUTS: None.

USE: This vector may be changed to intercept interrupts from an optional secondary communications interface.

INTERRUPT: C

LOCATION: 0000:0030

NAME: IRQ 4 - COMMUNICATIONS, PRIMARY

DESCRIPTION: Not used.

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

INPUTS: None.

OUTPUTS: None.

USE: This vector may be changed to intercept interrupts from an optional primary communications interface.

INTERRUPT: D

LOCATION: 0000:0034

NAME: IRQ 5

DESCRIPTION: Not used.

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

INPUTS: None.

OUTPUTS: None.

USE: If a user-supplied, interrupt-driven device uses the IRQ5 interrupt line, then location 0000:0034 is loaded with the address of the interrupt service routine.

INTERRUPT: E

LOCATION: 0000:0038

NAME: IRQ 6 - DISKETTE INTERRUPT

DESCRIPTION: This vector contains the address of the entry point of the diskette operation complete routine. The source of this interrupt is the diskette controller hardware.

INITIALIZED: Points to ROM Diskette Interrupt handler.

ROM ACTION: Calls made to the ROM using INT 13 for diskette I/O are suspended internally until INT E occurs, signifying completion.

INPUTS: None.

OUTPUTS: None.

USE: This vector may be changed to intercept printer character interrupts for special real-time operating system requirements.

INTERRUPT: F

LOCATION: 0000:003C

NAME: IRQ 7 - PRINTER INTERRUPT

DESCRIPTION: Not used.

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

INPUTS: None.

OUTPUTS: None.

USE: This vector may be changed to intercept printer character interrupts for special real-time operating system requirements.

INTERRUPT: 10h

LOCATION: 0000:0040

NAME: VIDEO I/O

DESCRIPTION: This software interrupt is called to perform all functions related to the video display.

INITIALIZED: Points to video I/O ROM entry point.

ROM ACTION: Upon entry, control is transferred to one of 16 (decimal) routines based on the function code in register AH. Illegal function codes cause control to simply return. All registers are preserved.

FUNCTION SUMMARY

AH	ACTION
0	SET MODE
1	SET CURSOR TYPE
2	SET CURSOR POSITION
3	READ CURSOR POSITION
4	READ LIGHT PEN POSITION
5	SELECT ACTIVE DISPLAY PAGE
6	SCROLL ACTIVE PAGE UP
7	SCROLL ACTIVE PAGE DOWN
8	READ ATTRIBUTE/CHARACTER
9	WRITE ATTRIBUTE/CHARACTER
10	WRITE CHARACTER ONLY
11	SET COLOR PALETTE
12	WRITE DOT
13	READ DOT
14	WRITE TTY
15	READ CURRENT VIDEO STATE

Interrupts remain enabled and execution may be suspended if CTRL+NUMLOCK is typed.

Functions and their related parameters are individually described following on pages 19 through 35.

VIDEO I/O (10h)
SET MODE

```
INPUTS:          AH = FUNCTION CODE = 0
                 AL = MODE (0..7)
                      AL = 0   40x25 B&W
                      AL = 1   40x25 Color
                      AL = 2   80x25 B&W
                      AL = 3   80x25 Color (default)
                      AL = 4   320x200 Color
                      AL = 5   320x200 B&W
                      AL = 6   640x200 B&W
                      AL = 7   80x25 External Monochrome

                    +---------------+---------------+
                 AH |       0       |     MODE      | AL
                    +---------------+---------------+
                 BX | /////////////////////////////// |
                    +-------------------------------+
                 CX | /////////////////////////////// |
                    +-------------------------------+
                 DX | /////////////////////////////// |
                    +-------------------------------+
```

OUTPUTS: None.

NOTES:

1. Modes 0 and 1 use 8x8 dot character cells from the character ROM on the VDU controller.
2. Modes 2 and 3 use the 9x14 dot character cells from the character ROM on the VDU controller. Alternately, the 8x8 dot mode can be chosen by pressing the CTRL, ALT, and < (less-than) keys simultaneously.
3. Modes 4, 5, and 6 display the 8x8 dot cell characters using ROM firmware to read a look-up table in the System ROM and write the appropriate dots on the screen.
4. Mode 7 directs video I/O to an external monochrome adapter card.
5. When switching between high-resolution (modes 2 or 3) and any other mode, there is a 500-ms delay to allow the internal monitor to change frequencies.
6. Color burst on the composite video output is not enabled in B&W modes; otherwise B&W and color operate in an identical manner.

VIDEO I/O (10h)
SET CURSOR TYPE

INPUTS:

AH = FUNCTION CODE = 1
CH = START LINE NUMBER FOR CURSOR IN BITS <4:0>
CL = END LINE NUMBER FOR CURSOR IN BITS <4:0>

```
   +---------------+---------------+
AH |       1       | ///////////// | AL
   +---------------+---------------+
BX | ///////////////////////////// |
   +---------------+---------------+
CH | START LINE #  |  END LINE #   | CL
   +---------------+---------------+
DX | ///////////////////////////// |
   +---------------+---------------+
```

OUTPUTS:

None.

NOTE:

The Set Cursor Type function takes special action if the current mode is 2 or 3 (80x25) and the high-resolution (9x14 dot cell) character set is in use. If the incoming start or stop line exceeds 7, the cursor is blanked. Otherwise, ROM code multiplies it by 14/8 and rounds the result to the nearest integer to map it to a cell that is actually 14 scan lines high, instead of 8.

The table below illustrates the adjustment:

IN	x 14/8	OUT (SCAN LINE)
0	0.00	0
1	1.75	2
2	3.50	4
3	5.25	5
4	7.00	7
5	8.75	9
6	10.50	11
7	12.25	12
8+	==BLANKED==	

VIDEO I/O (10h)
SET CURSOR POSITION

```
INPUTS:             AH = FUNCTION CODE = 2
                    BH = PAGE NUMBER (0..7) for Modes 0, 1;
                                     (0..3) for Modes 2, 3, 7
                    DH = ROW (0..24)
                    DL = COLUMN (0..39) for Modes 0, 1;
                                (0..79) for Modes 2, 3, 7

                       +---------------+---------------+
                    AH |       2       | ////////////// | AL
                       +---------------+---------------+
                    BH |  PAGE NUMBER  | ////////////// | BL
                       +---------------+---------------+
                    CX | /////////////////////////////// |
                       +---------------+---------------+
                    DH |      ROW      |    COLUMN     | DL
                       +---------------+---------------+

OUTPUTS:            None.
```

VIDEO I/O (10h)
READ CURSOR POSITION

```
INPUTS:          AH = FUNCTION CODE = 3
                 BH = PAGE NUMBER (0..7) for Modes 0, 1;
                                  (0..3) for Modes 2, 3, 7

                   +---------------+---------------+
                AH |       3       | ///////////// | AL
                   +---------------+---------------+
                BH |  PAGE NUMBER  | ///////////// | BL
                   +---------------+---------------+
                CX | ///////////////////////////// |
                   +-------------------------------+
                DX | ///////////////////////////// |
                   +-------------------------------+

OUTPUTS:         CH = START LINE NUMBER FOR CURSOR IN BITS <4:0>
                 CL = END LINE NUMBER FOR CURSOR IN BITS <4:0>
                 DH = ROW
                 DL = COLUMN

                   +-------------------------------+
                AX | ///////////////////////////// |
                   +-------------------------------+
                BX | ///////////////////////////// |
                   +---------------+---------------+
                CH | START LINE #  |  END LINE #   | CL
                   +---------------+---------------+
                DH |      ROW      |    COLUMN     | DL
                   +---------------+---------------+
```

NOTE: The Read Cursor Position function takes special action if the current mode is 2 or 3 (80x25) and the high-resolution (9x14 dot cell) character set is in use. Since the applications program expects a number in the range (0..7), and the physical line number is in the range (0..13), ROM code returns the line number by first multiplying it by 8/14 and then rounding the result to the nearest integer.

The table on the following page illustrates the mapping:

IN (SCAN LINE)	x 8/14	OUT (RETURNED)
0	0.00	0
1	0.57	1
2	1.14	1
3	1.71	2
4	2.29	2
5	2.86	3
6	3.43	3
7	4.00	4
8	4.57	5
9	5.14	5
10	5.71	6
11	6.28	6
12	6.86	7
13	7.43	7

VIDEO I/O (10h)
READ LIGHT PEN POSITION

```
INPUTS:            AH = FUNCTION CODE = 4

                     +---------------+---------------+
                  AH |       4       | ///////////// | AL
                     +---------------+---------------+
                  BX | ///////////////////////////// |
                     +-------------------------------+
                  CX | ///////////////////////////// |
                     +-------------------------------+
                  DX | ///////////////////////////// |
                     +-------------------------------+

OUTPUTS:           AH = 0 LIGHT PEN SWITCH NOT DOWN (not triggered)
                   AH = 1 VALID LIGHT PEN IN REGISTERS
                   BX = PIXEL COLUMN (0..319) for Modes 4, 5;
                                     (0..639) for Mode 6
                   CH = RASTER LINE (0..199)
                   DH = ROW OF CHARACTER LIGHT PEN IS ON
                   DL = COLUMN OF CHARACTER LIGHT PEN IS ON

                     +---------------+---------------+
                  AH |   PEN VALID   | ///////////// | AL
                     +---------------+---------------+
                  BX |          PIXEL COLUMN         |
                     +---------------+---------------+
                  CH |  RASTER LINE  | ///////////// | CL
                     +---------------+---------------+
                  DH |   CHAR ROW    |  CHAR COLUMN  | DL
                     +---------------+---------------+
```

NOTE: The light pen resolution is equivalent to character grid layout.

VIDEO I/O (10h)
SELECT ACTIVE DISPLAY PAGE FOR ALPHA MODES

```
INPUTS:            AH = FUNCTION CODE = 5
                   AL = NEW PAGE VALUE (0..7) for Modes 0, 1;
                                       (0..3) for Modes 2, 3, 7

                     +---------------+---------------+
                   AH |       5       |NEW PAGE VALUE | AL
                     +---------------+---------------+
                   BX | ///////////////////////////// |
                     +-------------------------------+
                   CX | ///////////////////////////// |
                     +-------------------------------+
                   DX | ///////////////////////////// |
                     +-------------------------------+
```

OUTPUTS: None.

NOTES: Invalid in graphics modes.

VIDEO I/O (10h)
SCROLL ACTIVE PAGE UP

INPUTS:

```
AH = FUNCTION CODE = 6
AL = NUMBER OF LINES TO SCROLL (0..25) for
     Modes  0, 1, 2, 3, 7.
     AL = 0 means blank entire window
BH = ATTRIBUTE TO BE USED ON BLANK LINES (00h..FFh)
CH = ROW OF UPPER-LEFT CORNER OF SCROLL (0..DH)
CL = COLUMN OF UPPER-LEFT CORNER OF SCROLL (0..DL)
DH = ROW OF LOWER-RIGHT CORNER OF SCROLL (CH..24)
DL = COLUMN OF LOWER-RIGHT CORNER OF SCROLL
     (CL..39) for Modes 0, 1;
     (CL..79) for Modes 2, 3, 7
```

```
                +---------------+---------------+
             AH |       6       |NUMBER OF LINES| AL
                +---------------+---------------+
       +-----BH |   ATTRIBUTE   | ///////////// | BL
       |        +---------------+---------------+
       |     CH |   U.L. ROW    |  U.L. COLUMN  | CL
       |        +---------------+---------------+
       |     DH |   L.R. ROW    |  L.R. COLUMN  | DL
       |        +---------------+---------------+
       |        +---+---+---+---+---+---+---+---+
       +--------| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | BH - FUNCTION
                +---+---+---+---+---+---+---+---+
                  |   |   |   |   |   |   |   |
                  |   |   |   |   |   |___|___|
                  |   |   |   |   |       |
                  |   |   |   |   |   FOREGROUND COLOR
                  |   |   |   |   |     000 = BLACK
                  |   |   |   |   |     001 = BLUE
                  |   |   |   |   |     010 = GREEN
                  |   |   |   |   |     011 = CYAN
                  |   |   |   |   |     100 = RED
                  |   |   |   |   |     101 = MAGENTA
                  |   |   |   |   |     110 = YELLOW
                  |   |   |   |   |     111 = WHITE
                  |   |   |   |   |
                  |   |   |   |   +-------INTENSE
                  |   |   |   |
                  |   +---+---+-------BACKGROUND COLOR
                  |                   (SELECTION AS ABOVE)
                  |
                  +-----------------------BLINK
```

OUTPUTS: None.

NOTES:

1. Input lines blanked at bottom of window.
2. The normal attribute has the value 07h.

VIDEO I/O (10h)
SCROLL ACTIVE PAGE DOWN

```
INPUTS:         AH = FUNCTION CODE = 7
                AL = NUMBER OF LINES TO SCROLL (0..25) for Modes
                     0, 1, 2, 3, 7.
                     AL = 0 means blank entire window
                BH = ATTRIBUTE TO BE USED ON BLANK LINES (00h..FFh)
                CH = ROW OF UPPER-LEFT CORNER OF SCROLL (0..DH)
                CL = COLUMN OF UPPER-LEFT CORNER OF SCROLL (0..DL)
                DH = ROW OF LOWER-RIGHT CORNER OF SCROLL (CH..24)
                DL = COLUMN OF LOWER-RIGHT CORNER OF SCROLL
                     (CL..39) for Modes 0, 1;
                     (CL..79) for Modes 2, 3, 7

                   +---------------+---------------+
                AH |       6       |NUMBER OF LINES| AL
                   +---------------+---------------+
          +-----BH |   ATTRIBUTE   | ///////////// | BL
          |        +---------------+---------------+
          |     CH |   U.L. ROW    |  U.L. COLUMN  | CL
          |        +---------------+---------------+
          |     DH |   L.R. ROW    |  L.R. COLUMN  | DL
          |        +---------------+---------------+
          |        +---+---+---+---+---+---+---+---+
          +--------| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | BH - FUNCTION
                   +---+---+---+---+---+---+---+---+
                     |   |   |   |   |   |   |   |
                     |   |   |   |   |   |___|___|
                     |   |   |   |   |       |
                     |   |   |   |   |   FOREGROUND COLOR
                     |   |   |   |   |     000 = BLACK
                     |   |   |   |   |     001 = BLUE
                     |   |   |   |   |     010 = GREEN
                     |   |   |   |   |     011 = CYAN
                     |   |   |   |   |     100 = RED
                     |   |   |   |   |     101 = MAGENTA
                     |   |   |   |   |     110 = YELLOW
                     |   |   |   |   |     111 = WHITE
                     |   |   |   |   |
                     |   |   |   |   +-------INTENSE
                     |   |   |   |
                     |   +---+---+-------BACKGROUND COLOR
                     |                   (SELECTION AS ABOVE)
                     |
                     +-----------------------BLINK
```

OUTPUTS: None.

NOTES:

1. Input lines blanked at top of window.
2. Normal attribute has the value 07h.

VIDEO I/O (10h)
READ ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION

```
INPUTS:          AH = FUNCTION CODE = 8
                 BH = DISPLAY PAGE IN ALPHA MODES ONLY
                      (0..7) for Modes 0, 1;
                      (0..3) for Modes 2, 3, 7

                   +---------------+---------------+
                 AH |       8       | ///////////// | AL
                   +---------------+---------------+
                 BH | DISPLAY PAGE  | ///////////// | BL
                   +---------------+---------------+
                 CX | ///////////////////////////// |
                   +-------------------------------+
                 DX | ///////////////////////////// |
                   +-------------------------------+

OUTPUTS:         AL = CHARACTER READ
                 AH = ATTRIBUTE OF CHARACTER READ

                   +---------------+---------------+
                 AH |   ATTRIBUTE   |   CHAR READ   | AL
                   +---------------+---------------+
                 BX | ///////////////////////////// |
                   +-------------------------------+
                 CX | ///////////////////////////// |
                   +-------------------------------+
                 DX | ///////////////////////////// |
                   +-------------------------------+
```

NOTES:

1. For read and write character functions while in graphics mode, the characters are formed from a character dot image in the System ROM. Only the first 128 characters are in the ROM. Interrupt vector 1Fh points to a user-supplied table that contains the dot patterns for the second 128 characters.

VIDEO I/O (10h)
WRITE ATTRIBUTE/CHARACTER AT CURRENT CURSOR POSITION

```
INPUTS:          AH = FUNCTION CODE = 9
                 AL = CHARACTER TO WRITE (00h..FFh)
                 BH = DISPLAY PAGE IN ALPHA MODES ONLY
                      (0..7) for Modes 0, 1;
                      (0..3) for Modes 2, 3, 7
                 BL = ATTRIBUTE OF CHARACTER TO WRITE
                      if bit <7> of AL = 1, then the color value
                      is XORed with the current contents of the
                      character.
                 CX = COUNT OF CHARACTERS TO WRITE
                      (1..1024) for Modes 0, 1;
                      (1..2048) for Modes 2, 3, 7;
                      (1..40) for Modes 4, 5;
                      (1..80) for Mode 6

                    +---------------+---------------+
                 AH |       9       |   CHARACTER   | AL
                    +---------------+---------------+
                 BH | DISPLAY PAGE  |   ATTRIBUTE   | BL
                    +---------------+---------------+
                 CX | COUNT OF CHARACTERS TO WRITE  |
                    +-------------------------------+
                 DX | ///////////////////////////// |
                    +-------------------------------+
```

OUTPUTS: None.

NOTES:

1. For read and write character functions while in graphics mode, the characters are formed from a character dot image in the System ROM. Only the first 128 characters are in the ROM. Interrupt vector 1Fh points to a user-supplied table that contains the dot patterns for the second 128 characters.
2. Maximum character count limited in alpha modes to end of display page.
3. When in modes 4, 5, or 6 (graphics mode), the replication factor contained in register CX will produce valid results only for characters on the same row.
4. Composition of the attribute byte is shown on page 26.

VIDEO I/O (10h)
WRITE CHARACTER ONLY AT CURRENT CURSOR POSITION

```
INPUTS:            AH = FUNCTION CODE = 10
                   AL = CHARACTER TO WRITE
                   BH = DISPLAY PAGE IN ALPHA MODES ONLY
                        (0..7) for Modes 0, 1;
                        (0..3) for Modes 2, 3, 7
                   CX = COUNT OF CHARACTERS TO WRITE
                        (1..1024) for Modes 0, 1;
                        (1..2048) for Modes 2, 3, 7;
                        (1..40) for Modes 4, 5;
                        (1..80) for Mode 6
```

```
   +---------------+---------------+
AH |      10       |   CHARACTER   | AL
   +---------------+---------------+
BH | DISPLAY PAGE  | ///////////// | BL
   +---------------+---------------+
CX | COUNT OF CHARACTERS TO WRITE  |
   +-------------------------------+
DX | ///////////////////////////// |
   +-------------------------------+
```

OUTPUTS: None.

NOTES:

1. For read and write character functions while in graphics mode, the characters are formed from a character dot image in the System ROM. Only the first 128 characters are in the ROM. Interrupt vector 1Fh points to a user-supplied table that contains the dot patterns for the second 128 characters.
2. Maximum character count is limited in alpha modes to end of display page.
3. When in modes 4, 5, or 6 (graphics mode), the replication factor contained in register CX will produce valid results only for characters on the same row.

VIDEO I/O (10h)
SET COLOR PALETTE

```
INPUTS:         AH = FUNCTION CODE = 11
                BH = PALETTE COLOR ID BEING SET (0..127)
                BL = COLOR VALUE TO BE USED WITH THAT COLOR ID
                     This entry point has meaning only for Modes 4, 5
                     Color  ID = 0 selects the  background  color
                                 (0..15)
                     Color ID = 1 selects the palette to be used
```

```
   +---------------+---------------+
AH |      11       | ///////////// | AL
   +---------------+---------------+
BH |   COLOR ID    |  COLOR VALUE  | BL
   +---------------+---------------+
CX | ///////////////////////////// |
   +-------------------------------+
DX | ///////////////////////////// |
   +-------------------------------+
```

VALUE	COLOR
0	BLACK
1	BLUE
2	GREEN
3	CYAN
4	RED
5	MAGENTA
6	BROWN
7	WHITE
8	GRAY
9	LIGHT BLUE
10	LIGHT GREEN
11	LIGHT CYAN
12	LIGHT RED
13	LIGHT MAGENTA
14	YELLOW
15	WHITE

OUTPUTS: None.

VIDEO I/O (10h)
WRITE DOT

```
INPUTS:          AH = FUNCTION CODE = 12
                 AL = COLOR VALUE
                      (0..3) or (80h..83h) for Modes 4, 5;
                      (0..1) or (80h..81h) for Mode 6
                      if bit <7> of AL is set to 1, then the color
                      value is XORed  with the current contents of
                      the dot.
                 CX = COLUMN NUMBER
                      (0..319) for Modes 4, 5;
                      (0..639) for Mode 6
                 DX = ROW NUMBER (0..199)

                    +---------------+---------------+
                 AH |      12       |  COLOR VALUE  | AL
                    +---------------+---------------+
                 BX | /////////////////////////////// |
                    +-------------------------------+
                 CX |         COLUMN NUMBER         |
                    +-------------------------------+
                 DX |          ROW NUMBER           |
                    +-------------------------------+
```

VALUE	SET	COLOR
0	0	BACKGROUND
1	0	GREEN
2	0	RED
3	0	YELLOW
0	1	BACKGROUND
1	1	CYAN
2	1	MAGENTA
3	1	WHITE

OUTPUTS: None.

VIDEO I/O (10h)
READ DOT

```
INPUTS:          AH = FUNCTION CODE = 13
                 CX = COLUMN NUMBER
                      (0..319) for Modes 4, 5;
                      (0..639) for Mode 6
                 DX = ROW NUMBER (0..199)

                    +---------------+---------------+
                 AH |       13      | ///////////// | AL
                    +---------------+---------------+
                 BX | ///////////////////////////// |
                    +-------------------------------+
                 CX |         COLUMN NUMBER         |
                    +-------------------------------+
                 DX |           ROW NUMBER          |
                    +-------------------------------+

OUTPUTS:         AL = THE DOT READ
                      The Dot Read has value (0..3) for Modes 4, 5;
                      and the value (0..1) for Mode 6.

                    +---------------+---------------+
                 AH | ///////////// |   DOT VALUE   | AL
                    +---------------+---------------+
                 BX | ///////////////////////////// |
                    +-------------------------------+
                 CX | ///////////////////////////// |
                    +-------------------------------+
                 DX | ///////////////////////////// |
                    +-------------------------------+

NOTE:            This  function has significance only for Modes 4,
                 5, and 6 (graphics modes).
```

VIDEO I/O (10h)
WRITE TTY

```
INPUTS:             AH = FUNCTION CODE = 14
                    AL = CHAR TO WRITE
                    BH = DISPLAY PAGE IN ALPHA MODES ONLY
                          (0..7) for Modes 0, 1;
                          (0..3) for Modes 2, 3, 7
                    BL = FOREGROUND COLOR IN GRAPHICS MODE (0..3) for
                                                   Modes 4, 5;
                                                   (0..1) for Mode 6

                       +---------------+---------------+
                    AH |      14       |   CHARACTER   | AL
                       +---------------+---------------+
                    BH | DISPLAY PAGE  | FOREGND COLOR | BL
                       +---------------+---------------+
                    CX | ///////////////////////////// |
                       +-------------------------------+
                    DX | ///////////////////////////// |
                       +-------------------------------+
```

OUTPUTS: None.

NOTE: Screen width is controlled by the previously set mode.

VIDEO I/O (10h)
READ CURRENT VIDEO STATE

INPUTS: AH = FUNCTION CODE = 15

```
    +---------------+---------------+
 AH |      15       | ///////////// | AL
    +---------------+---------------+
 BX | ///////////////////////////// |
    +-------------------------------+
 CX | ///////////////////////////// |
    +-------------------------------+
 DX | ///////////////////////////// |
    +-------------------------------+
```

OUTPUTS: AH = NUMBER OF CHARACTER COLUMNS ON
                        SCREEN (40 or 80)
                   AL = MODE CURRENTLY SET (0..7)
                   BH = CURRENT ACTIVE DISPLAY PAGE (0..7)

```
    +---------------+---------------+
 AH | SCREEN WIDTH  | CURRENT MODE  | AL
    +---------------+---------------+
 BH |  ACTIVE PAGE  | ///////////// | BL
    +---------------+---------------+
 CX | ///////////////////////////// |
    +-------------------------------+
 DX | ///////////////////////////// |
    +-------------------------------+
```

INTERRUPT: 11h

LOCATION: 0000:0044

NAME: EQUIPMENT CONFIGURATION

DESCRIPTION: Returns to the caller a bit-encoded word relating the number and type of hardware devices installed.

INITIALIZED: Points to Equipment Configuration ROM entry point.

ROM ACTION: The equipment status word is initialized on power up. Calls by INT 11h return the contents of this word in register AX.

INPUTS: None.

OUTPUTS: AX = EQUIPMENT CONFIGURATION

```
   +---------------------------------+
AX |     EQUIPMENT CONFIGURATION     |
   +---------------------------------+
BX | /////////////////////////////// |
   +---------------------------------+
CX | /////////////////////////////// |
   +---------------------------------+
DX | /////////////////////////////// |
   +---------------------------------+
```

EQUIPMENT CONFIGURATION (continued)

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|15|14|13|12|11|10| 9| 8| 7| 6| 5| 4| 3| 2| 1| 0| AX - FUNCTION
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  +-- BOOTED FROM DISK
 |  |  |  |  |  |  |  |  |  |  |  |  |  |  +----- 8087 INSTALLED
 |  |  |  |  |  |  |  |  |  |  |  |  +--+-------- PLANAR RAM SIZE
 |  |  |  |  |  |  |  |  |  |  |  |                 00 = 16K
 |  |  |  |  |  |  |  |  |  |  |  |                 01 = 32K
 |  |  |  |  |  |  |  |  |  |  |  |                 10 = 48K
 |  |  |  |  |  |  |  |  |  |  |  |                 11 = 64K
 |  |  |  |  |  |  |  |  |  |  +--+----------- INITIAL VIDEO CONFIG
 |  |  |  |  |  |  |  |  |  |                    00 = UNUSED
 |  |  |  |  |  |  |  |  |  |                    01 = INT. 40x25
 |  |  |  |  |  |  |  |  |  |                    10 = INT. 80x25
 |  |  |  |  |  |  |  |  |  |                    11 = EXT. MONOCHR
 |  |  |  |  |  |  |  |  +--+----------------- NUMBER OF DISK DRIVES
 |  |  |  |  |  |  |  |                          00 = 1 DRIVE
 |  |  |  |  |  |  |  |                          01 = 2 DRIVES
 |  |  |  |  |  |  |  |                          10 = 3 DRIVES
 |  |  |  |  |  |  |  |                          11 = 4 DRIVES
 |  |  |  |  |  |  |  +----------------------- RESERVED
 |  |  |  |  +--+--+--+----------------------- NUMBER RS-232 PORTS
 |  |  |  |                                      000 = NO PORTS
 |  |  |  |                                      001 = 1 PORT
 |  |  |  |                                      010 = 2 PORTS
 |  |  |  |                                      011 = 3 PORTS
 |  |  |  |                                      100 = 4 PORTS
 |  |  |  |                                      101 = 5 PORTS
 |  |  |  |                                      110 = 6 PORTS
 |  |  |  |                                      111 = 7 PORTS
 |  |  |  +----------------------------------- GAME I/O ADAPTER
 |  |  +-------------------------------------- RESERVED
 +--+----------------------------------------- NUMBER OF PRINTERS
                                                 00 = NO PRINTERS
                                                 01 = 1 PRINTER
                                                 10 = 2 PRINTERS
                                                 11 = 3 PRINTERS
```

NOTES:

1. Bit <0> must always be a 1 (always booted from disk).
2. Bits <3:2> must always indicate 64K planar memory.
3. Bits <5:4> normally indicate an initial video configuration of internal monitor, 80x25.

INTERRUPT: 12h

LOCATION: 0000:0048

NAME: MEMORY SIZE

DESCRIPTION: Returns to the caller the amount of contiguous RAM installed in 1-Kbyte increments.

INITIALIZED: Points to Memory Size ROM entry point.

ROM ACTION: The memory size word is initialized on power-up. Calls by INT 12h return the contents of this word in AX. It contains the number of contiguous 1-Kbyte blocks of memory.

INPUTS: None.

OUTPUTS: AX = CONTIGUOUS MEMORY SIZE IN 1-KBYTE INCREMENTS

```
   +--------------------------------+
AX |          MEMORY SIZE           |
   +--------------------------------+
BX | ////////////////////////////// |
   +--------------------------------+
CX | ////////////////////////////// |
   +--------------------------------+
DX | ////////////////////////////// |
   +--------------------------------+
```

NOTES:

1. Minimum RAM configuration on the COMPAQ Computer is 128 Kbytes.
2. The number in AX is always in multiples of 32 (decimal), since it reflects the RAM switch settings.

INTERRUPT: 13h

LOCATION: 0000:004C

NAME: DISKETTE I/O

DESCRIPTION: This software interrupt is called to perform all functions related to diskette I/O.

INITIALIZED: Points to Diskette I/O ROM entry point.

ROM ACTION: Upon entry, control is transferred to one of six routines based on the function code in register AH. Illegal function codes cause control to simply return. All registers except those returning a value are preserved.

FUNCTION SUMMARY

AH	ACTION
0	RESET DISK DRIVE SYSTEM
1	SENSE STATUS
2	READ SECTORS
3	WRITE SECTORS
4	VERIFY
5	FORMAT TRACK

Interrupts remain enabled.

Functions and their related parameters are individually described following on pages 40 through 46.

NOTES:

1. Limits on disk sector parameters refer to MS-DOS™ only.
2. Diskette hardware supports only two drives (0..1).
3. CF is the 8088 CPU carry flag.
4. Other operating parameters may be effected by changing the Diskette Parameter Table (see Interrupt 1Eh on page 70).

DISKETTE I/O (13h)
RESET DISK DRIVE SYSTEM

```
INPUTS:          AH = FUNCTION CODE = 0

                    +---------------+---------------+
                 AH |       0       | ///////////// | AL
                    +---------------+---------------+
                 BX | ///////////////////////////// |
                    +-------------------------------+
                 CX | ///////////////////////////// |
                    +-------------------------------+
                 DX | ///////////////////////////// |
                    +-------------------------------+

OUTPUTS         AH = STATUS
                    AH = 00h   Successful completion.
                    AH = 80h   Hardware fault.
                AL = 0
                CF = STATUS
                    CF = 0   Successful completion.
                    CF = 1   Hardware fault.

                    +---------------+---------------+
                 AH |    STATUS     |       0       | AL
                    +---------------+---------------+
                 BX | ///////////////////////////// |
                    +-------------------------------+
                 CX | ///////////////////////////// |
                    +-------------------------------+
                 DX | ///////////////////////////// |
                    +-------------------------------+

NOTES:           Also includes controller.
```

DISKETTE I/O (13h)
SENSE STATUS

```
INPUTS:          AH = FUNCTION CODE = 1

                    +---------------+---------------+
                 AH |       1       | ///////////// | AL
                    +---------------+---------------+
                 BX | ///////////////////////////// |
                    +-------------------------------+
                 CX | ///////////////////////////// |
                    +-------------------------------+
                 DX | ///////////////////////////// |
                    +-------------------------------+

OUTPUTS:         AH = STATUS
                     AH = 80h   Device failed to respond.
                     AH = 40h   Seek operation failed.
                     AH = 20h   LSI controller failed.
                     AH = 10h   Bad CRC on diskette read.
                     AH = 09h   Attempt to perform DMA across 64-K
                                boundary.
                     AH = 08h   DMA overrun on operation.
                     AH = 04h   Requested sector not found.
                     AH = 03h   Attempt to write on protected
                                diskette.
                     AH = 02h   Address mark not found.
                     AH = 01h   Bad command.
                     AH = 00h   Success.
                 AL = 0
                 CF = 0

                    +---------------+---------------+
                 AH |    STATUS     |       0       | AL
                    +---------------+---------------+
                 BX | ///////////////////////////// |
                    +-------------------------------+
                 CX | ///////////////////////////// |
                    +-------------------------------+
                 DX | ///////////////////////////// |
                    +-------------------------------+
```

NOTES:

1. Status sensed is from last operation.
2. CF is always cleared.

DISKETTE I/O (13h)
READ SECTORS

```
INPUTS:          AH = FUNCTION CODE = 2
                 AL = SECTORS COUNT (1..8)
                 CH = TRACK NUMBER (0..39)
                 CL = SECTOR NUMBER (1..8)
                 DH = HEAD NUMBER (0..1)
                 DL = DRIVE NUMBER (0..3)
                 ES:BX = ADDRESS OF BUFFER

                    +---------------+---------------+
                 AH |       2       | SECTOR COUNT  | AL
                    +---------------+---------------+
                 BX |     BUFFER ADDRESS OFFSET     |
                    +---------------+---------------+
                 CH |     TRACK     |    SECTOR     | CL
                    +---------------+---------------+
                 DH |     HEAD      |     DRIVE     | DL
                    +---------------+---------------+

                    +-------------------------------+
                 ES |    BUFFER ADDRESS SEGMENT     |
                    +-------------------------------+

OUTPUTS:         AH = STATUS
                     AH = 80h   Device failed to respond.
                     AH = 40h   Seek operation failed.
                     AH = 20h   LSI controller failed.
                     AH = 10h   Bad CRC on diskette read.
                     AH = 09h   Attempt to perform DMA across 64-K
                                boundary.
                     AH = 08h   DMA overrun on operation.
                     AH = 04h   Requested sector not found.
                     AH = 02h   Address mark not found.
                     AH = 01h   Bad command.
                     AH = 00h   Success.
                 AL = 0
                 CF = 0 FOR OK; 1 FOR ERROR

                    +---------------+---------------+
                 AH |    STATUS     |       0       | AL
                    +---------------+---------------+
                 BX | ///////////////////////////// |
                    +-------------------------------+
                 CX | ///////////////////////////// |
                    +-------------------------------+
                 DX | ///////////////////////////// |
                    +-------------------------------+
```

DISKETTE I/O (13h)
WRITE SECTORS

```
INPUTS:          AH = FUNCTION CODE = 3
                 AL = SECTORS COUNT (1..8)
                 CH = TRACK NUMBER (0..39)
                 CL = SECTOR NUMBER (1..8)
                 DH = HEAD NUMBER (0..1)
                 DL = DRIVE NUMBER (0..3)
                 ES:BX = ADDRESS OF BUFFER

                    +---------------+---------------+
                 AH |       3       | SECTOR COUNT  | AL
                    +---------------+---------------+
                 BX |     BUFFER ADDRESS OFFSET     |
                    +---------------+---------------+
                 CH |     TRACK     |    SECTOR     | CL
                    +---------------+---------------+
                 DH |     HEAD      |     DRIVE     | DL
                    +---------------+---------------+

                    +-------------------------------+
                 ES |    BUFFER ADDRESS SEGMENT     |
                    +-------------------------------+

OUTPUTS:         AH = STATUS
                     AH = 80h   Device failed to respond.
                     AH = 40h   Seek operation failed.
                     AH = 20h   LSI controller failed.
                     AH = 10h   Bad CRC on diskette read.
                     AH = 09h   Attempt to perform DMA across 64-K
                                boundary.
                     AH = 08h   DMA overrun on operation.
                     AH = 04h   Requested sector not found.
                     AH = 03h   Attempt to write on protected diskette.
                     AH = 02h   Address mark not found.
                     AH = 01h   Bad command.
                     AH = 00h   Success.
                 AL = 0
                 CF = 0 FOR OK; 1 FOR ERROR

                    +---------------+---------------+
                 AH |    STATUS     |       0       | AL
                    +---------------+---------------+
                 BX | ///////////////////////////// |
                    +-------------------------------+
                 CX | ///////////////////////////// |
                    +-------------------------------+
                 DX | ///////////////////////////// |
                    +-------------------------------+
```

DISKETTE I/O (13h)
VERIFY

```
INPUTS:          AH = FUNCTION CODE = 4
                 AL = SECTORS COUNT (1..8)
                 CH = TRACK NUMBER (0..39)
                 CL = SECTOR NUMBER (1..8)
                 DH = HEAD NUMBER (0..1)
                 DL = DRIVE NUMBER (0..3)

                   +---------------+---------------+
                 AH |       4       | SECTOR COUNT  | AL
                   +---------------+---------------+
                 BX | ///////////////////////////// |
                   +---------------+---------------+
                 CH |     TRACK     |    SECTOR     | CL
                   +---------------+---------------+
                 DH |     HEAD      |     DRIVE     | DL
                   +---------------+---------------+

OUTPUTS:         AH = STATUS
                     AH = 80h   Device failed to respond.
                     AH = 40h   Seek operation failed.
                     AH = 20h   LSI controller failed.
                     AH = 10h   Bad CRC on diskette read.
                     AH = 09h   Attempt to perform DMA across 64-K
                                boundary.
                     AH = 08h   DMA overrun on operation.
                     AH = 04h   Requested sector not found.
                     AH = 02h   Address mark not found.
                     AH = 01h   Bad command.
                     AH = 00h   Success.
                 AL = 0
                 CF = 0 FOR OK; 1 FOR ERROR

                   +---------------+---------------+
                 AH |    STATUS     |       0       | AL
                   +---------------+---------------+
                 BX | ///////////////////////////// |
                   +-------------------------------+
                 CX | ///////////////////////////// |
                   +-------------------------------+
                 DX | ///////////////////////////// |
                   +-------------------------------+
```

DISKETTE I/O (13h)
FORMAT TRACK

```
INPUTS:          AH = FUNCTION CODE = 5
                 AL = SECTORS COUNT
                 CH = TRACK NUMBER (0..39)
                 DH = HEAD NUMBER (0..1)
                 DL = DRIVE NUMBER (0..3)
                 ES:BX = ADDRESS OF BUFFER

                    +---------------+---------------+
                 AH |       5       | SECTOR COUNT  | AL
                    +---------------+---------------+
                 BX |      TABLE POINTER OFFSET     |
                    +---------------+---------------+
                 CH |     TRACK     | ////////////  | CL
                    +---------------+---------------+
                 DH |     HEAD      |     DRIVE     | DL
                    +---------------+---------------+

                    +-------------------------------+
                 ES |     TABLE POINTER SEGMENT     |
                    +-------------------------------+

                    FORMAT TABLE (FIRST ENTRY)

                      +---------------+
                 +00h |  TRACK        |  (0..39)
                      +---------------+
                 +01h |  HEAD         |  (0..1)
                      +---------------+
                 +02h |  SECTOR       |  (1..s)
                      +---------------+
                 +03h |      N        |  (0..3) N = 0 128 bytes
                      +---------------+         N = 1 256 bytes
                              .                 N = 2 512 bytes
                              .                 N = 3 1024 bytes
                              .

OUTPUTS:         AH = STATUS
                      AH = 80h   Device failed to respond.
                      AH = 40h   Seek operation failed.
                      AH = 20h   LSI controller failed.
                      AH = 10h   Bad CRC on diskette read.
                      AH = 09h   Attempt to perform DMA across 64-K
                                 boundary.
                      AH = 08h   DMA overrun on operation.
                      AH = 04h   Requested sector not found.
                      AH = 02h   Address mark not found.
                      AH = 01h   Bad command.
                      AH = 00h   Success.
                 AL = 0
                 CF = 0 FOR OK; 1 FOR ANY ERROR
```

```
   +---------------+---------------+
AH |    STATUS     |       0       | AL
   +---------------+---------------+
BX | ///////////////////////////// |
   +-------------------------------+
CX | ///////////////////////////// |
   +-------------------------------+
DX | ///////////////////////////// |
   +-------------------------------+
```

NOTES:

1. When using the format call, the table pointer (ES:BX) must point to a table of 4-byte entries which give the desired address fields used to format the diskette.

INTERRUPT: 14h

LOCATION: 0000:0050

NAME: COMMUNICATIONS I/O

DESCRIPTION: This software interrupt is called to perform all functions related to I/O on the RS-232 serial ports.

INITIALIZED: Points to Communications I/O ROM entry point.

ROM ACTION: Upon entry, control is transferred to one of four routines based on the function code in register AH. Illegal function codes cause control to simply return. All registers except those returning a value are preserved.

FUNCTION SUMMARY

AH	ACTION
0	INITIALIZE PORT
1	TRANSMIT CHARACTER
2	RECEIVE CHARACTER
3	SENSE STATUS

Interrupts remain enabled.

Functions and their related parameters are individually described following on pages 48 through 51.

COMMUNICATIONS I/O (14h)
INITIALIZE PORT

```
INPUTS:         AH = FUNCTION CODE = 0
                DX = COMMUNICATIONS ADAPTER SELECT (0..1)

                   +---------------+---------------+
                AH |       0       | CONFIGURATION | AL ---+
                   +---------------+---------------+       |
                BX | ///////////////////////////// |       |
                   +-------------------------------+       |
                CX | ///////////////////////////// |       |
                   +-------------------------------+       |
                DX |         ADAPTER SELECT        |       |
                   +-------------------------------+       |
                                                           |
                                                           |
        +---+---+---+---+---+---+---+---+                  |
        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | AL - FUNCTION ----+
        +---+---+---+---+---+---+---+---+
          |   |   |   |   |   |   |   |
          |   |   |   |   |   |   +---+---- WORD LENGTH
          |   |   |   |   |   |               10 = 7 BITS
          |   |   |   |   |   |               11 = 8 BITS
          |   |   |   |   |   +------------ STOP BITS
          |   |   |   |   |                   0 = 1 BIT
          |   |   |   |   |                   1 = 2 BITS
          |   |   |   +---+---------------- PARITY
          |   |   |                           x0 = NONE
          |   |   |                           01 = ODD
          |   |   |                           11 = EVEN
          +---+---+------------------------ BAUD RATE
                                              000 =  110 BAUD
                                              001 =  150 BAUD
                                              010 =  300 BAUD
                                              011 =  600 BAUD
                                              100 = 1200 BAUD
                                              101 = 2400 BAUD
                                              110 = 4800 BAUD
                                              111 = 9600 BAUD

OUTPUTS:        None.
```

```
                         COMMUNICATIONS I/O (14h)
                         TRANSMIT CHARACTER IN AL

INPUTS:           AH = FUNCTION CODE = 1
                  AL = CHARACTER TO BE TRANSMITTED (00h..FFh)
                  DX = COMMUNICATIONS ADAPTER SELECT (0..1)

                     +--------------+--------------+
                  AH |      1       |   CHARACTER  | AL
                     +--------------+--------------+
                  BX | ///////////////////////////// |
                     +-----------------------------+
                  CX | ///////////////////////////// |
                     +-----------------------------+
                  DX |        ADAPTER SELECT       |
                     +-----------------------------+

OUTPUTS:          AH = STATUS
                       Bit <7> of AH = 1 if error

                     +--------------+--------------+
                  AH |    STATUS    | ///////////// | AL
                     +--------------+--------------+
                  BX | ///////////////////////////// |
                     +-----------------------------+
                  CX | ///////////////////////////// |
                     +-----------------------------+
                  DX | ///////////////////////////// |
                     +-----------------------------+
```

COMMUNICATIONS I/O (14h)
RECEIVE CHARACTER

```
INPUTS:        AH = FUNCTION CODE = 2
               DX = COMMUNICATIONS ADAPTER SELECT (0..1)

                   +--------------+--------------+
                AH |      2       | ///////////// | AL
                   +--------------+--------------+
                BX | //////////////////////////// |
                   +-----------------------------+
                CX | //////////////////////////// |
                   +-----------------------------+
                DX |        ADAPTER SELECT        |
                   +-----------------------------+

OUTPUTS:       AH = STATUS
                    AH nonzero if error
               AL = CHARACTER RECEIVED

                   +--------------+--------------+
                AH |    STATUS    |  CHARACTER   | AL
                   +--------------+--------------+
                BX | //////////////////////////// |
                   +-----------------------------+
                CX | //////////////////////////// |
                   +-----------------------------+
                DX | //////////////////////////// |
                   +-----------------------------+
```

COMMUNICATIONS I/O (14h)
SENSE COMMUNICATIONS STATUS

```
INPUTS:         AH = FUNCTION CODE = 3
                DX = COMMUNICATIONS ADAPTER SELECT (0..1)

                   +---------------+----------------+
                AH |       3       | ////////////// | AL
                   +---------------+----------------+
                BX | ////////////////////////////// |
                   +--------------------------------+
                CX | ////////////////////////////// |
                   +--------------------------------+
                DX |         ADAPTER SELECT         |
                   +--------------------------------+

OUTPUTS:        AH = UART STATUS
                AL = MODEM LINE STATUS

                   +---------------+----------------+
  +------------ AH |  UART STATUS  |  LINE STATUS   | AL --+
  |                +---------------+----------------+      |
  |             BX | ////////////////////////////// |      |
  |                +--------------------------------+      |
  |             CX | ////////////////////////////// |      |
  |                +--------------------------------+      |
  |             DX | ////////////////////////////// |      |
  |                +--------------------------------+      |
  |                                                        |
  |   +---+---+---+---+---+---+---+---+                    |
  |   | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | AL - FUNCTION -----+
  |   +---+---+---+---+---+---+---+---+
  |     |   |   |   |   |   |   |   +---- DELTA CLEAR-TO-SEND
  |     |   |   |   |   |   |   +-------- DELTA DATA-SET-READY
  |     |   |   |   |   |   +------------ TRAILING EDGE RING
  |     |   |   |   |   |                   INDICATOR
  |     |   |   |   |   +---------------- DELTA RX LINE SIGNAL
  |     |   |   |   |                       DETECT
  |     |   |   |   +-------------------- CLEAR-TO-SEND
  |     |   |   +------------------------ DATA-SET-READY
  |     |   +---------------------------- RING INDICATOR
  |     +-------------------------------- RX LINE SIGNAL DETECT
  |
  |     +---+---+---+---+---+---+---+---+
  +---  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | AH - FUNCTION
        +---+---+---+---+---+---+---+---+
          |   |   |   |   |   |   |   +-- RX DATA READY
          |   |   |   |   |   |   +------ RX OVERRUN ERROR
          |   |   |   |   |   +---------- RX PARITY ERROR
          |   |   |   |   +-------------- RX FRAMING ERROR
          |   |   |   +------------------ RX BREAK DETECT
          |   |   +---------------------- TX HOLDING REGISTER EMPTY
          |   +-------------------------- TX SHIFT REGISTER EMPTY
          +------------------------------ TIMEOUT
```

INTERRUPT: 15h

LOCATION:	0000:0054
NAME:	--
DESCRIPTION:	Not used.
INITIALIZED:	Points to an Interrupt Return.
ROM ACTION:	Returns.

INTERRUPT: 16h

LOCATION: 0000:0058

NAME: KEYBOARD I/O

DESCRIPTION: This software interrupt is called to perform all functions related to keyboard I/O.

INITIALIZED: Points to Keyboard I/O ROM entry point.

ROM ACTION: Upon entry, control is transferred to one of three routines based on the function code in register AH. Illegal function codes cause control to simply return. All registers except those returning a value are preserved.

FUNCTION SUMMARY

AH	ACTION
0	GET KEY
1	CHECK FOR KEY AVAILABLE
2	READ SHIFT STATUS

Interrupts remain enabled.

Functions and their related parameters are individually described following on pages 54 through 56.

KEYBOARD I/O (16h)
GET KEY

```
INPUTS:            AH = FUNCTION CODE = 0

                      +---------------+---------------+
                   AH |       0       | ///////////// | AL
                      +---------------+---------------+
                   BX | ///////////////////////////// |
                      +-------------------------------+
                   CX | ///////////////////////////// |
                      +-------------------------------+
                   DX | ///////////////////////////// |
                      +-------------------------------+

OUTPUTS:           AH = SCAN CODE (01h..53h)
                   AL = CHARACTER (00h..FFh)

                      +---------------+---------------+
                   AH |   SCAN CODE   |   CHARACTER   | AL
                      +---------------+---------------+
                   BX | ///////////////////////////// |
                      +-------------------------------+
                   CX | ///////////////////////////// |
                      +-------------------------------+
                   DX | ///////////////////////////// |
                      +-------------------------------+
```

KEYBOARD I/O (16h)
CHECK FOR KEY AVAILABLE

```
INPUTS:           AH = FUNCTION CODE = 1

                     +---------------+---------------+
                  AH |       1       | ///////////// | AL
                     +---------------+---------------+
                  BX | /////////////////////////////// |
                     +-------------------------------+
                  CX | /////////////////////////////// |
                     +-------------------------------+
                  DX | /////////////////////////////// |
                     +-------------------------------+

OUTPUTS:          ZF = RESULT
                          ZF = 1 means no keys available
                          ZF = 0 means AH = SCAN CODE,
                                   and AL = CHARACTER
                  AH = SCAN CODE (01h..53h)
                  AL = CHARACTER (00h..FFh)

                     +---------------+---------------+
                  AH |   SCAN CODE   |   CHARACTER   | AL
                     +---------------+---------------+
                  BX | /////////////////////////////// |
                     +-------------------------------+
                  CX | /////////////////////////////// |
                     +-------------------------------+
                  DX | /////////////////////////////// |
                     +-------------------------------+

NOTES:          1. The  character is not removed from  the  keyboard
                   input buffer.
                2. ZF is the 8088 CPU zero flag.
```

```
                          KEYBOARD I/O (16h)
                           READ SHIFT STATUS

INPUTS:            AH = FUNCTION CODE = 2

                      +---------------+---------------+
                   AH |       2       | ///////////// | AL
                      +---------------+---------------+
                   BX | ///////////////////////////// |
                      +-------------------------------+
                   CX | ///////////////////////////// |
                      +-------------------------------+
                   DX | ///////////////////////////// |
                      +-------------------------------+

OUTPUTS:           AL = SHIFT CODES

                      +---------------+---------------+
                   AH | ///////////// |  SHIFT CODES  | AL --+
                      +---------------+---------------+      |
                   BX | ///////////////////////////// |      |
                      +-------------------------------+      |
                   CX | ///////////////////////////// |      |
                      +-------------------------------+      |
                   DX | ///////////////////////////// |      |
                      +-------------------------------+      |
                                                             |
        +---+---+---+---+---+---+---+---+                    |
        | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | AL - FUNCTION -----+
        +---+---+---+---+---+---+---+---+
          |   |   |   |   |   |   |   |
          |   |   |   |   |   |   |   +---- RIGHT SHIFT KEY HELD DOWN
          |   |   |   |   |   |   +-------- LEFT SHIFT KEY HELD DOWN
          |   |   |   |   |   +------------ CTRL KEY HELD DOWN
          |   |   |   |   +---------------- ALT KEY HELD DOWN
          |   |   |   +-------------------- SCROLL LOCK  KEY  WAS PRESSED
          |   |   +------------------------ NUM LOCK KEY WAS PRESSED
          |   +---------------------------- CAPS LOCK KEY WAS PRESSED
          +-------------------------------- INS KEY WAS PRESSED
```

INTERRUPT: 17h

LOCATION: 0000:005C

NAME: PRINTER I/O

DESCRIPTION: This software interrupt is called to perform all functions related to printer I/O.

INITIALIZED: Points to Printer I/O ROM entry point.

ROM ACTION: Upon entry, control is transferred to one of three routines based on the function code in register AH. Illegal function codes cause control to simply return. All registers except those returning a value are preserved.

FUNCTION SUMMARY

AH	ACTION
0	PRINT CHARACTER
1	INITIALIZE
2	GET STATUS

Interrupts remain enabled.

Functions and their related parameters are individually described following on pages 58 through 60.

PRINTER I/O (17h)
PRINT CHARACTER

```
INPUTS:           AH = FUNCTION CODE = 0
                  AL = CHARACTER TO BE PRINTED (00h..FFh)
                  DX = ADAPTER SELECT NUMBER (0..2)

                     +---------------+---------------+
                  AH |       0       |   CHARACTER   | AL
                     +---------------+---------------+
                  BX | ///////////////////////////// |
                     +-------------------------------+
                  CX | ///////////////////////////// |
                     +-------------------------------+
                  DX |        ADAPTER SELECT         |
                     +-------------------------------+

OUTPUTS:          AH = STATUS

                     +---------------+---------------+
              /   AH |    STATUS     | ///////////// | AL
              |      +---------------+---------------+
              |   BX | ///////////////////////////// |
              |      +-------------------------------+
              |   CX | ///////////////////////////// |
              |      +-------------------------------+
              |   DX | ///////////////////////////// |
              |      +-------------------------------+
              |      +---+---+---+---+---+---+---+---+
              +--->  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | AH - STATUS
                     +---+---+---+---+---+---+---+---+
                       |   |   |   |   |   |   |   |
                       |   |   |   |   |   |   |   +---- TIME OUT
                       |   |   |   |   |   +---+-------- 0 0
                       |   |   |   |   +---------------- I/O ERROR
                       |   |   |   +-------------------- SELECTED
                       |   |   +------------------------ OUT OF PAPER
                       |   +---------------------------- ACKNOWLEDGE
                       +-------------------------------- READY
```

PRINTER I/O (17h)
INITIALIZE

```
INPUTS:            AH = FUNCTION CODE = 1
                   DX = ADAPTER SELECT NUMBER (0..2)

                      +---------------+---------------+
                   AH |       1       | ///////////// | AL
                      +---------------+---------------+
                   BX | ///////////////////////////// |
                      +-------------------------------+
                   CX | ///////////////////////////// |
                      +-------------------------------+
                   DX |        ADAPTER SELECT         |
                      +-------------------------------+

OUTPUTS:           AH = STATUS

                      +---------------+---------------+
                /  AH |    STATUS     | ///////////// | AL
                |     +---------------+---------------+
                |  BX | ///////////////////////////// |
                |     +-------------------------------+
                |  CX | ///////////////////////////// |
                |     +-------------------------------+
                |  DX | ///////////////////////////// |
                |     +-------------------------------+
                |     +---+---+---+---+---+---+---+---+
                +---> | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | AH - STATUS
                      +---+---+---+---+---+---+---+---+
                        |   |   |   |   |   |   |   |
                        |   |   |   |   |   |   |   +---- TIME OUT
                        |   |   |   |   |   +---+-------- 0 0
                        |   |   |   |   +---------------- I/O ERROR
                        |   |   |   +-------------------- SELECTED
                        |   |   +------------------------ OUT OF PAPER
                        |   +---------------------------- ACKNOWLEDGE
                        +-------------------------------- READY
```

PRINTER I/O (17h)
GET STATUS

```
INPUTS:            AH = FUNCTION CODE = 2
                   DX = ADAPTER SELECT NUMBER (0..2)

                      +---------------+---------------+
                   AH |       2       | ///////////// | AL
                      +---------------+---------------+
                   BX | ///////////////////////////// |
                      +-------------------------------+
                   CX | ///////////////////////////// |
                      +-------------------------------+
                   DX |        ADAPTER SELECT         |
                      +-------------------------------+

OUTPUTS:           AH = STATUS
                        AH, bit <0> = 0 means success

                      +---------------+---------------+
                /  AH |    STATUS     | ///////////// | AL
                |     +---------------+---------------+
                |  BX | ///////////////////////////// |
                |     +-------------------------------+
                |  CX | ///////////////////////////// |
                |     +-------------------------------+
                |  DX | ///////////////////////////// |
                |     +-------------------------------+
                |
                |     +---+---+---+---+---+---+---+---+
                +---> | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | AH - STATUS
                      +---+---+---+---+---+---+---+---+
                        |   |   |   |   |   |   |   |
                        |   |   |   |   |   |   |   +---- TIME OUT
                        |   |   |   |   |   +---+-------- 0 0
                        |   |   |   |   +---------------- I/O ERROR
                        |   |   |   +-------------------- SELECTED
                        |   |   +------------------------ OUT OF PAPER
                        |   +---------------------------- ACKNOWLEDGE
                        +-------------------------------- READY
```

INTERRUPT: 18h

LOCATION:	0000:0060
NAME:	--
DESCRIPTION:	Not used.
INITIALIZED:	0000:0000
ROM ACTION:	None.

INTERRUPT: 19h

LOCATION: 0000:0064

NAME: BOOTSTRAP

DESCRIPTION: Reads the bootstrap loader from the system disk and transfers control to it.

INITIALIZED: Points to Bootstrap ROM entry point.

ROM ACTION: Enables interrupts. Note that although most INT routines do this upon entry, this is the first time system interrupts are allowed from power-up or reboot. Three attempts are then made to read the boot sector into location 0000:7C00. If these fail, the messages "Non-System disk or disk error", "Replace and strike any key when ready" are printed, prompting the user to insert a good system disk into Drive A:.

INPUTS: None.

OUTPUTS: None.

INTERRUPT: 1Ah

LOCATION: 0000:0068

NAME: TIME OF DAY I/O

DESCRIPTION: Reads or sets the time-of-day as updated by the interrupting clock.

INITIALIZED: Points to Time-of-Day I/O ROM entry point.

ROM ACTION: Upon entry, control is transferred to one of two routines based on the function code in register AH. Illegal function codes cause control to simply return. All registers except those returning a value are preserved.

FUNCTION SUMMARY

AH	ACTION
0	READ T-O-D
1	SET T-O-D

Interrupts remain enabled.

Functions and their related parameters are individually described following on pages 64 and 65.

TIME-OF-DAY (1Ah)
READ T-O-D

```
INPUTS:             AH = FUNCTION CODE = 0

                       +---------------+---------------+
                    AH |       0       | ///////////// | AL
                       +---------------+---------------+
                    BX | ///////////////////////////// |
                       +-------------------------------+
                    CX | ///////////////////////////// |
                       +-------------------------------+
                    DX | ///////////////////////////// |
                       +-------------------------------+

OUTPUTS:            CX = HIGH PORTION OF COUNT
                    DX = LOW PORTION OF COUNT
                    AL = ROLLED OVER FLAG
                         AL = 1 if 24 hours have elapsed since last read

                       +---------------+---------------+
                    AH | ///////////// |  ROLLED OVER  | AL
                       +---------------+---------------+
                    BX | ///////////////////////////// |
                       +-------------------------------+
                    CX |        COUNT M.S. WORD        |
                       +-------------------------------+
                    DX |        COUNT L.S. WORD        |
                       +-------------------------------+
```

TIME-OF-DAY (1Ah)
SET T-O-D

INPUTS: AH = FUNCTION CODE = 1
CX = HIGH PORTION OF COUNT
DX = LOW PORTION OF COUNT

```
   +---------------+---------------+
AH |       1       | ///////////// | AL
   +---------------+---------------+
BX | ///////////////////////////// |
   +-------------------------------+
CX |        COUNT M.S. WORD        |
   +-------------------------------+
DX |        COUNT L.S. WORD        |
   +-------------------------------+
```

OUTPUTS: None.

NOTES: The ROLLED OVER FLAG is cleared.

INTERRUPT: 1Bh

LOCATION: 0000:006C

NAME: CTRL-BREAK VECTOR

DESCRIPTION: This interrupt is called from the ROM when the CTRL+BREAK key combination is struck. It is provided to allow DOS and user programs a way to asynchronously exit a program.

INITIALIZED: Points to Dummy Interrupt Return.

ROM ACTION: Returns.

USE: This vector is normally used by the operating system. It can be changed to point to a user-supplied routine.

INTERRUPT: 1Ch

LOCATION:	0000:0070
NAME:	TIMER VECTOR
DESCRIPTION:	This interrupt is called from the ROM every hardware clock tick to provide the user with a means of executing code on a periodic basis (18.2 times per second).
INITIALIZED:	Points to Dummy Interrupt Return.
ROM ACTION:	Returns.
INPUTS:	None.
OUTPUTS:	None.
USE:	It can be changed to point to a user-supplied routine to be called at each hardware timer clock tick (18.2 times per second). The user routine must save all registers used and return with IRET.

INTERRUPT: 1Dh

LOCATION: 0000:0074

NAME: VIDEO INIT PARAMETER TABLE VECTOR

DESCRIPTION: Points to an optional user-supplied table to initialize the 6845 CRT controller after an INT 10h MODE command is issued.

INITIALIZED: Points to ROM Default Video Init Parameter Table.

ROM ACTION: Not applicable.

INPUTS: None.

OUTPUTS: None.

USE: Can be used to supply a substitute parameter table for different CRT controller operating conditions. For example, the start of the active display area can be changed to compensate for certain monitors.

The vector at 0000:0074 can be changed from the table in ROM to point to a user-supplied one consisting of four 16-byte entries.

```
Offset +-------------------------+
 +00h  |      40x25 Table        |
       |       Modes 0, 1        |
       +-------------------------+
 +10h  |      80x25 Table        |
       |       Modes 2, 3        |
       +-------------------------+
 +20h  |     Graphics Table      |
       |      Modes 4, 5, 6      |
       +-------------------------+
 +30h  |    Monochrome Table     |
       |         Mode 7          |
       +-------------------------+
```

There are actually two such sets of tables in ROM to accomodate switching between 9x14 dot cells and 8x8 dot cells in the 80x25 character mode. Switching is done by firmware in the ROM inside the keyboard interrupt handler. The CTRL+ALT+< (less-than) forces the 80x25 display to 8x8 dot cell characters and the complementary CTRL+ALT+> (greater-than) switches to 9x14 dot cells. Note that only the table for the 80x25 entry changes from 8x8 mode to 9x14 mode.
The format of the tables is as follows:

'SET		DEFAULTS (Decimal)							
		9x14 dot				8x8 dot			
		40x25	80x25	Graph	Mono	40x25	80x25	Graph	Mono
+00h	HORIZ TOTAL IN CHARS	56	113	56	97	56	113	56	97
+01h	HORIZ DISPL IN CHARS	40	80	40	80	40	80	40	80
+02h	HORIZ SYNC POSN CHARS	45	90	45	82	45	90	45	82
+03h	HORIZ SYNC WIDTH CHARS	10	10	10	15	10	10	10	15
+04h	VERT TOTAL IN CHAR ROWS	31	25	127	25	31	31	127	25
+05h	VERT TOTAL ADJ IN SCAN	6	6	6	6	6	6	6	6
+06h	VERT DISPL IN CHAR ROWS	25	25	100	25	25	25	100	25
+07h	VERT SYNC POSN CHAR ROW	28	25	112	25	28	28	112	25
+08h	INTERLACE MODE	2	2	2	2	2	2	2	2
+09h	MAX SCAN LINE ADDRESS	7	13	1	13	7	7	1	13
+0Ah	CURSOR START SCAN	6	11	6	11	6	6	6	11
+0Bh	CURSOR END SCAN	7	12	7	12	7	7	7	12
:h	START ADDRESS (H)	0	0	0	0	0	0	0	0
+0Dh	START ADDRESS (L)	0	0	0	0	0	0	0	0
+0Eh	CURSOR ADDRESS (H)	0	0	0	0	0	0	0	0
+0Fh	CURSOR ADDRESS (L)	0	0	0	0	0	0	0	0

INTERRUPT: 1Eh

LOCATION: 0000:0078

NAME: DISKETTE PARAMETER TABLE VECTOR

DESCRIPTION: This vector points to a diskette parameter table used to program the diskette controller chip (NEC 765 or Intel 8272).

The table is organized as follows:

OFFSET	7	6	5	4	3	2	1	0	DEFAULT	COMMENTS
+00h	STEP RATE				HEAD UNLOAD				DFh	SRT = 1..16 ms (F=1,E=2,etc); HUT = 16..240 in 16 ms incr
+01h	HEAD LOAD TIME							DMA	02h	HLT = 2..254 in 2 ms incr; DMA = 0 for DMA mode
+02h	MOTOR OFF TIME (/55 ms)								25h	Wait time in 18.2 Hz ticks; Motor Off Time is 2 s.
+03h	N								02h	N = 0, 1, 2, 3 for Sector length = 128, 256, 512, 1024
+04h	SECTORS/TRACK								08h	
+05h	GAP LENGTH, NORMAL								2Ah	
06h	DTL								FFh	Or sector length if N = 0
+07h	GAP LENGTH, FORMAT								50h	Used by Format command
+08h	FILL CHARACTER, FORMAT								F6h	Used by Format command
+09h	HEAD SETTLE TIME (/1 ms)								00h	
+0Ah	MOTOR SETTLE TIME (/125 ms)								04h	Wait 500 ms for motor to come up to speed.

Refer to either the NEC 765 or Intel 8272 disk controller specification for more information.

INITIALIZED: Points to ROM Default Diskette Parameter Table.

ROM ACTION: Not applicable.

INPUTS: See above.

OUTPUTS: None.

USE: May be used to read/write other diskette formats.

INTERRUPT: 1Fh

LOCATION: 0000:007C

NAME: GRAPHICS DOT TABLE VECTOR

DESCRIPTION: This vector points to a user-supplied dot table used to generate and read 8x8 dot graphics characters in modes 4, 5, and 6. This table is needed only for those characters within the range of 128..255 (decimal).

INITIALIZED: 0000:0000

ROM ACTION: It is used exclusively by the INT 5 print screen and INT 10h video I/O routines, and then only in the three graphics modes for the upper-128 character set.

INPUTS: None.

OUTPUTS: None.

USE: It is the responsibility of the user to load this vector pointing to a supplied table. The table is of the form:

(continued on next page)

```
              +--------------------------- Left-most column
              |
              |                   +------ Right-most column
              |                   |
byte offset +--+--+--+--+--+--+--+--+
    +000h   |  |  |  |  |  |  |  |  | Top row     \
            +--+--+--+--+--+--+--+--+              |
    +001h   |  |  |  |  |  |  |  |  |              |
            +--+--+--+--+--+--+--+--+              |
    +002h   |  |  |  |  |  |  |  |  |              |
            +--+--+--+--+--+--+--+--+              |
    +003h   |  |  |  |  |  |  |  |  |               \
            +--+--+--+--+--+--+--+--+                } Char 128
    +004h   |  |  |  |  |  |  |  |  |               /
            +--+--+--+--+--+--+--+--+              |
    +005h   |  |  |  |  |  |  |  |  |              |
            +--+--+--+--+--+--+--+--+              |
    +006h   |  |  |  |  |  |  |  |  |              |
            +--+--+--+--+--+--+--+--+              |
    +007h   |  |  |  |  |  |  |  |  | Bottom row  /
            +--+--+--+--+--+--+--+--+
                        :
            +--+--+--+--+--+--+--+--+
    +3F8h   |  |  |  |  |  |  |  |  | Top row     \
            +--+--+--+--+--+--+--+--+              |
    +3F9h   |  |  |  |  |  |  |  |  |               \
            +--+--+--+--+--+--+--+--+                } Char 255
                        :                           /
            +--+--+--+--+--+--+--+--+              |
    +3FFh   |  |  |  |  |  |  |  |  | Bottom row  /
            +--+--+--+--+--+--+--+--+
```

TABLE ENTRIES FOR 8x8 DOT CHARACTER CELLS

Make QUESTION MARK (?) the character for code 81h.

Begin at offset +008h. (81h - 80h) * 8 = +008h. Left-justify alphanumeric characters in the cell. Visible dots are usually composed of two adjacent cells ON. Bottom row is normally blank (00h), except for descenders and special graphics characters.

```
OFFSET    +---+---+---+---+---+---+---+---+
 +008h    |   | * | * | * | * |   |   |   |  78h
          +---+---+---+---+---+---+---+---+
 +009h    | * | * |   |   | * | * |   |   |  CCh
          +---+---+---+---+---+---+---+---+
 +00Ah    |   |   |   |   | * | * |   |   |  0Ch
          +---+---+---+---+---+---+---+---+
 +00Bh    |   |   |   | * | * |   |   |   |  18h
          +---+---+---+---+---+---+---+---+
 +00Ch    |   |   | * | * |   |   |   |   |  30h
          +---+---+---+---+---+---+---+---+
 +00Dh    |   |   |   |   |   |   |   |   |  00h
          +---+---+---+---+---+---+---+---+
 +00Eh    |   |   | * | * |   |   |   |   |  30h
          +---+---+---+---+---+---+---+---+
 +00Fh    |   |   |   |   |   |   |   |   |  00h
          +---+---+---+---+---+---+---+---+
```

SPECIAL ROM LOCATION

ATION: F000:FFE6

NAME: REVISION

DESCRIPTION: ROM revision in ASCII, left-justified, blank-filled.

USE: This 4-byte location may be read to determine the current ROM revision level. The contents are a single upper-case letter, preceded or followed by zero to three asterisks (*). Unused positions to the right are filled with blanks (20h).

NOTE: This location has this meaning only on the COMPAQ Computer. See MACHINE ID on page 75.

SPECIAL ROM LOCATION

ATION: F000:FFEA

NAME: MACHINE ID

DESCRIPTION: Unit Identifier.

USE: This 6-byte location may be read to determine if the applications program is running on a COMPAQ Computer. It contains the letters `COMPAQ' all in upper-case ASCII. Some applications programs may want to take advantage of a COMPAQ-specific feature. For example, the screen memory can be scrolled faster, since it is fast enough to be read and written without first turning off the video or waiting for the vertical retrace interval.